

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Veronika Blažič

Aplikacija za učenje tipkanja za slepe in slabovidne

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Matija Marolt

Ljubljana 2015

Fakulteta za računalništvo in informatiko podpira javno dostopnost znanstvenih, strokovnih in razvojnih rezultatov. Zato priporoča objavo dela pod katero od licenc, ki omogočajo prosto razširjanje diplomskega dela in/ali možnost nadaljne proste uporabe dela. Ena izmed možnosti je izdaja diplomskega dela pod katero od Creative Commons licenc <http://creativecommons.si>

Morebitno pripadajočo programsko kodo praviloma objavite pod, denimo, licenco *GNU General Public License*, različica 3. Podrobnosti licence so dostopne na spletni strani <http://www.gnu.org/licenses/>.

Besedilo je oblikovano z urejevalnikom besedil L^AT_EX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V okviru diplomske naloge izdelajte spletno aplikacijo za učenje desetprstnega tipkanja, ki bo prilagojena otrokom in še posebej dostopna slepi in slabovidni mladini. Preučite obstoječe podobne aplikacije in v sodelovanju z Zavodom za slepo in slabovidno mladino določite glavne funkcionalne zahteve, ki jih v okviru naloge tudi realizirajte.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisana Veronika Blažič sem avtorica diplomskega dela z naslovom:

Aplikacija za učenje tipkanja za slepe in slabovidne

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelala samostojno pod mentorstvom doc. dr. Matije Marolta,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 5. septembra 2015

Podpis avtorja:

*Zahvaljujem se svojemu mentorju, doc. dr. Matiji Maroltu, za vso pomoč
tekom razvoja in pisanja diplomske naloge, ter Matevžu Pesku za vse nasvete
in usmerjanje. Hvala tudi vsem ostalim sodelujočim na projektu Zaznaj-
Spoznaj, predvsem Matevžu in Piji, brez katerih bi končni izdelek izgledal in
deloval pol tako dobro, ter ekipi ZSSM za njihove ideje in napotke.*

*Rada bi se zahvalila svojim staršem za njihovo moralno, strokovno in fi-
nančno podporo tekom študija.*

Matija, hvala za vso podporo in pomoč pri prvih poskusih programiranja.

*Hvala tudi vsem sošolcem in prijateljem, ki so mi v času študija kakorkoli
pomagali, predvsem pa Neži in Nives, ki sta študij naredili neizmerno prije-
tnejši.*

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Slepo desetprstno tipkanje	3
2.1	Pravilna drža	3
2.2	Začetna postavitev rok	5
2.3	Pravila desetprstnega tipkanja	5
2.4	Pozitivne posledice znanja slepega tipkanja	6
3	Razvoj aplikacij za slepe in slabovidne	9
3.1	Slabovidnost	9
3.2	Pripomočki za branje zaslona	11
3.3	Principi razvoja aplikacij za slepe in slabovidne	13
4	Pregled področja	17
4.1	TypingClub	17
4.2	keybr.com	19
4.3	Stamina	20
5	Aplikacija za učenje tipkanja	23
5.1	Zahteve za aplikacijo	23
5.2	Tehnologije	24

KAZALO

5.3 Rešitev	31
6 Sklepne ugotovitve	39
Literatura	41

Seznam uporabljenih kratic

kratica	angleško	slovensko
W3C	World Wide Web Consortium	Konzorcij svetovnega spleta
SVG	Scalable Vector Graphics	skalabilna vektorska grafika
DOM	The Document Object Model	Model dokument/objekt
AJAX	asynchronous JavaScript and XML	asinhronski Javascript in XML
UI	user interface	uporabniški vmesnik
WAI-ARIA	Web Accessibility Initiative – Accessible Rich Internet Applications	pobuda k spletni dostopnosti - dostopne in bogate internetne aplikacije
AFB	American Foundation for the Blind	Ameriški sklad za slepe
ZSSM	Association for the Blind and Partially Sighted Youth	Zveza za slepo in slabovidno mladino

Povzetek

V informacijski dobi, v kateri se trenutno nahajamo, je hitro tipkanje ena najpomembnejših veščin, ki jih potrebujemo za učinkovito uporabo računalnika. V sklopu te diplomske naloge smo razvili aplikacijo, ki uporabnika vodi na poti do znanja slepega desetprstnega tipkanja, ki lahko opazno poviša posameznikovo hitrost tipkanja. Sodelovali smo tudi z Zavodom za slepo in slabovidno mladino v Ljubljani, ki obenem deluje kot izobraževalna ustanova, ter našo rešitev prilagodili za uporabo njihovih učencev tako, da smo izdelali prilagodljiv vmesnik, ki ga je moč popolnoma personalizirati.

Ključne besede: slepo, desetprstno, tipkanje.

Abstract

Fast typing is nowadays one of the most important skills needed to efficiently use a computer. As a part of this thesis, we have developed an application, that guides its user towards mastering touch typing, which can significantly increase one's typing speed. We collaborated with the Association for the Blind and Partially Sighted Youth in Ljubljana, which is also an educational institution, and adapted our solution to suit their student's needs by creating a customizable interface that can be completely personalized.

Keywords: touch, typing.

Poglavje 1

Uvod

Računalniki so v zadnjih nekaj desetletjih postali nepogrešljiv del vsakega gospodinjstva. Njihovo uporabo se predpostavlja tako v učnih ustanovah, kot tudi na večini delovnih mest, zato je skoraj nujno, da vsak posameznik obvlada njegovo osnovno uporabo.

Slepo tipkanje je veščina, ki drastično poveča hitrost in učinkovitost uporabe računalnika. Število natipkanih besed na minuto je lahko pri ljudeh, ki to veščino obvladajo, tudi dvakrat večje kot pri navadnih uporabnikih. Poleg tega se je pri slepem tipkanju lažje osredtočiti na dogajanje na zaslonu, saj ni potrebe po konstantnem preklapljanju pogleda med zaslonom in tipkovnico, kar viša tudi produktivnost.

Za primarno povratno informacijo računalnik uporablja zaslon. V tem pogledu so osebe, ki imajo z vidom težave, nekoliko na slabšem, zato so bili razviti različni pripomočki, ki olajšajo branje z zaslona. Mednje sodijo Braillovi prikazovalniki, bralci in povečevalci zaslona. Kljub tem pripomočkom je priporočljivo tudi znanje desetprstnega tipkanja, saj lahko potem te osebe tipkajo, ne da bi jim bilo treba iskati črke po tipkovnici, obenem pa jim je lažje slediti dogajanju na zaslonu, če jim ni treba neprestano preklapljati pogleda med zaslonom in tipkovnico.

Za diplomsko nalogo smo izdelali spletno aplikacijo, ki bo njenim uporab-

nikom pomagala na poti osvajanja veščine desetprstnega tipkanja. Pri sami izdelavi smo sodelovali tudi z Zavodom za slepo in slabovidno mladino v Ljubljani, ki deluje tudi kot izobraževalna ustanova, katere osnovnošolci se morajo naučiti slepega desetprstnega tipkanja kot del učnega programa. Pri učenju tipkanja si pomagajo z različnimi namiznimi programi, ki pa imajo vsak svoje prednosti in slabosti. Čeprav se ti programi prilagodijo obliki tipkovnice, ki jo uporablja operacijski sistem, so povečini v angleščini, nudijo pa ne nobenih osnovnih navodil in morajo te učitelji pripraviti posebej. Z izdelavo spletne aplikacije smo želeli doseči predvsem to, da bodo vsa gradiva in naloge za tipkanje dostopni na istem mestu, poleg tega pa bo imel učitelj pregled nad napredkom celotnega razreda, saj bo aplikacija beležila tudi napredek učencev skozi čas, česar večina obstoječih aplikacij ne vključuje, ker ne hranijo posameznih rezultatov. Spletne aplikacije so tudi veliko bolj prenosljive, saj za svoje delovanje potrebujejo le spletni brskalnik in dostop do interneta, kar bo učencem omogočilo lažje vaje tipkanja tudi doma. Pri razvoju naše aplikacije smo se osredotočili na to, da je vmesnik popolnoma prilagodljiv in ima lahko posamezen uporabnik svojo instanco barv, uporabili smo razločno in lahko berljivo pisavo ter prilagodili vmesnik za uporabo povečevalcev zaslona, saj vsi ti naštetí dejavniki pomagajo ljudem z okvarami vida pri lažji uporabi računalnika.

Diplomska naloga je razdeljena na štiri dele. V prvem si bomo ogledali pravila slepega desetprstnega tipkanja, v drugem delu bomo govorili o razvoju aplikacij za slepe in slabovidne, v tretjem bomo analizirali nekaj podobnih rešitev, v četrtem delu pa si bomo ogledali tehnologije in metodologije, ki jih bomo uporabili, ter analizirali našo rešitev.

Poglavje 2

Slepo desetprstno tipkanje

Dandanes je skoraj vsakdo primoran uporabljati tipkovnico. Seveda je mogoče tipkati z uporabo dveh ali treh prstov, vendar je tako tipkanje zamudno in natipkano besedilo polno napak. Veliko boljši način je tipkanje z uporabo vseh desetih prstov, pri čemer se naučimo, katera črka pripada kateremu prstu, podobno kot pri igranju inštrumenta.

Desetprstno slepo tipkanje je veščina, ki se je lahko naučimo v nekaj urah, z nekaj vaje pa nam lahko za vedno olajša uporabo tipkovnice. Vešči slepega tipkanja namreč tipkajo veliko hitreje, poleg tega pa naredijo ob tipkanju manj napak in so, ker jim ni treba neprestano iskati črk po tipkovnici, tudi veliko bolj produktivni.

Pravila desetprstnega slepega tipkanja se od tipkovnice do tipkovnice nekoliko razlikujejo. V nadaljevanju si bomo ogledali pravila za slovensko tipkovnico QWERTZ, ki se uporablja na večini slovenskih računalnikov.

2.1 Pravilna drža

Poleg slepega tipkanja se skoraj vedno omenja tudi pravilno držo, saj poleg večje hitrosti tipkanja in manjšega števila napak slepo tipkanje uči tudi ostale prakse zdrave uporabe računalnika. Naslednji odstavki so povzeti po [6].

Ob veliki uporabi računalnika in nepravilnem sedenju za pisalno mizo lahko čutimo, kako to neugodno vpliva na naše telo. Najpogosteje se pojavijo bolečine v ramenih, vratu in hrbtu, ob tipkanju pa tudi v zapestjih in prstih. Če se naučimo pravilne drže, pa ostanejo mišice sproščene in se počasneje utrudimo.

V teoriji je najprimenejši vzravnani položaj. Ta položaj narekuje vzravnano hrbtenico, komolci pa naj držijo, ko so roke na tipkovnici, pravi kot. Zapestji naj bosta čim bolj zravnani. Hrbet naj bo naslonjen na stolno naslonjalo in vrat čim bolj zravnani. Kot med telesom in nogami naj bo večji ali enak 90 stopinjam, stopali naj bosta na tleh. Kot v kolenskem zgibu naj bo med 90 in 110 stopinjami, zgornja dela rok ob telesu. Zaslon naj bo v višini oči in od njih oddaljen med 40 in 70 cm. Tipkovnica naj bo od roba mize oddaljena toliko, da lahko nanj naslonimo zapestja.



Slika 2.1: Primer pravilnega in napačnega sedenja [7]

Poglavji 2.2 in 2.3 povzeti po [3].

2.2 Začetna postavitve rok

Ob začetku desetprstnega slepega tipkanja prste postavimo v izhodiščni položaj. Levi kazalec naj počiva na črki F, desni pa na črki J. Ostale prste postavimo na sosednje tipke, tako da je levi mezinček na črki A, prstanec na S in sredinec na D, desni mezinček pa na črki Č, prstanec na L in sredinec na K. Enega od palcev (če smo desničarji navadno desnega, če smo levičarji navadno levega) uporabljamo za pritisk na preslednico, drugi pa ne počne nič. Lahko bi rekli, da gre v resnici za devetprstno tipkanje.

2.3 Pravila desetprstnega tipkanja

Ko imamo prste v izhodiščnem položaju, lahko ostale tipke dosežemo s pomikanjem prstov gor in dol, mezinca pa se uporabljata še za pritisk tipk desno oziroma levo od njiju. Pomembno je, da se vsi prsti po vsakem natipkanem znaku vrnejo v svojo izhodiščno lego. Spodaj so za posamezen prst našteje tipke, ki jih doseže [3].



Slika 2.2: Obseg posameznih prstov

Levi mezinček: levi mezinček se uporablja za pritisk tipk A, Q, < in Y ter tudi vseh ostalih, ki ležijo levo od teh, kot so escape, tabulator, caps lock, shift in levi ctrl. Uporablja se tudi za pritisk 1 in 2, do znakov dostopnih s pritiskom

na shift pa se dostopa tako, da z desnim mezincem pritisnemo na desni shift.

Levi prstanec: levi prstanec se uporablja za pritisk tipk S, W in X ter 3. Za doseg znakov, za katere se rabi shift, se uporablja desna shift tipka.

Levi sredinec: levi sredinec se uporablja za pritisk tipk D, E in C ter 4. Za doseg znakov, za katere se rabi shift, se uporablja desna shift tipka.

Levi kazalec: levi kazalec se uporablja za pritisk tipk F, R, V, T, G in B ter 5 in 6. Za doseg znakov, za katere se rabi shift, se uporablja desna shift tipka.

Desni kazalec: desni kazalec se uporablja za pritisk tipk H, Z, N, U, J in M ter 7 in 8. Za doseg znakov, za katere se rabi shift, se uporablja leva shift tipka.

Desni sredinec: desni sredinec se uporablja za pritisk tipk K, I in , ter 9. Za doseg znakov, za katere se rabi shift, se uporablja leva shift tipka.

Desni prstanec: desni prstanec se uporablja za pritisk tipk L, O in . ter 0. Za doseg znakov, za katere se rabi shift, se uporablja leva shift tipka.

Desni mezinec: desni mezinec se uporablja za pritisk tipk Č, P, -, ' in + ter preostalih tipk, ki se nahajajo desno od njegovega izhodišča kot so desni shift, enter in backspace. Za doseg znakov, za katere se rabi shift, se uporablja leva shift tipka.

2.4 Pozitivne posledice znanja slepega tipkanja

O pozitivnih posledicah znanja desetprstnega slepega tipkanja je nekaj napisanega že v prejšnjih poglavjih. Spodaj je nekoliko podrobneje razčlenjenih

osem, povzetih po [8].

1. **Hitrost:** je najbolj očitna posledica znanja desetprstnega tipkanja. Vešči desetprstnega tipkanja lahko tipkajo tudi trikrat hitreje od običajnih uporabnikov tipkovnice. Na hitrost ne vpliva le to, da imajo v glavi ustvarjeno povezavo med črkami in prsti, temveč tudi to, da jim ni treba po tipkovnici iskati črk in preklapljati pogleda med zaslonom ter tipkovnico.
2. **Natančnost:** je ena najpomembnejših posledic znanja desetprstnega tipkanja. Napake večjih desetprstnega tipkanja so minimalne, ne glede na to, kako hitro tipkajo.
3. **Čas:** je posledica hitrosti. Ker se z znanjem slepega tipkanja poveča hitrost, se namreč posledično premo sorazmerno zmanjša čas, ki je potreben za natipkanje nekega besedila. Ker se hitrost tipkanja lahko potroji, se lahko tudi zmanjša čas tipkanja za trikrat.
4. **Utrujenost:** Če tipkamo dalj časa, nas to utruje tako fizično kot tudi psihično. Veščina slepega tipkanja zmanjša obe utrujenosti. Psihično je tovrstna oblika tipkanja manj zahtevna, ker se nam ni treba osredotočati na dve stvari naenkrat. Osredotočimo se tako le na rezultat na zaslonu, ne pa tudi na tipke na tipkovnici. Fizično manj zahtevno pa je slepo tipkanje zato, ker k njegovemu znanju praviloma sodi tudi znanje o pravilni drži pri sedenju za računalnikom in imamo tako posledično praviloma manj težav z bolečinami v hrbtu in rokah ter bolj sproščene mišice.
5. **Zdravje:** Ker je slepo tipkanje manj psihično in fizično zahtevno je tudi manj stresno. Poleg tega imajo ljudje, ki znajo desetprstno tipkati, manj težav z bolečinami v hrbtu in rokah, ker se načeloma držijo predpisov o zdravem sedenju za računalnikom ali pa so vsaj seznanjeni z njimi.

6. **Zaposljivost:** Znanje tipkanja je dandanes potrebno na večini delovnih mest. Če se lahko ob iskanju zaposlitve pohvališ z znanjem zelo hitrega tipkanja, boš najbrž veliko lažje dobil delovno mesto kot kandidati, ki hitrega tipkanja ne znajo.
7. **Osredotočanje:** Z znanjem desetprstnega tipkanja se lahko osredotočimo le na dogajanje na zaslonu in nam tako ni treba stalno preklapljati pozornosti med tipkovnico in zaslonom. Ker se nam ni treba osredotočati na dve stvari naenkrat, se nam zviša produktivnost, poleg tega pa lahko več pozornosti namenimo podrobnostim, ki bi jih sicer spregledali.
8. **Končne napake:** Če vso svojo pozornost namenimo zaslonu, bomo veliko verjetneje že med tipkanjem opazili tipkarske napake in jih sproti popravili. Tako bo v natipkanem besedilu na koncu tipkanja veliko manj napak in bo verjetnost, da ob končnem urejanju katero spregledamo, veliko manjša.

Poglavje 3

Razvoj aplikacij za slepe in slabovidne

Pri uporabi računalnika moramo nenehno opravljati dve stvari: prebirati informacijo, ki nam jo sporoča računalnik, navadno jo vidimo na zaslonu, in vnašati podatke ter ukaze v računalnik s pomočjo miške ali tipkovnice.

Pri vnašanju ukazov v računalnik slabovidni načeloma nimajo težav. Nekateri sicer ne morejo uporabljati miške, ker je kurzor majhen in se stalno premika ter ga posledično težje vidijo, a nima nihče težav pri uporabi tipkovnice, sploh če je več slepega tipkanja. Težava se pojavi pri prejemanju povratne informacije od računalnika, ker so podatki na zaslonu zanje težje vidni. Velja torej, da je treba pri razvoju aplikacij za slepe in slabovidne poskrbeti, da so informacije dostopne tudi zvočno ali na Braillovih prikazovalnikih, aplikacija pa mora podpirati tudi uporabo tipkovnice in ne le izključno miške.

3.1 Slabovidnost

Slabovidnost je zmanjšana sposobnost vida do te mere, da se je ne da izboljšati z raznimi pripomočki, kot so na primer očala ali kontaktne leče.

Poznamo več oblik slabovidnosti, ki so posledice različnih okvar očesa. Najpogostejše so napake pri lomljenju žarkov v očesu, med katere sodita tudi kratkovidnost ter daljnovidnost, in predstavljajo 43% oblik slabovidnosti. Sledi jim siva mrena, bolezen očesa pri kateri pride do zameglitve očesne leče, zaradi česar pride v oko manj svetlobe, in predstavlja 33% vseh slabovidnosti. Simptomi sive mreže so med drugim tudi poslabšan barvni vid in kontrast. Opazna oblika slabovidnosti je še glaukom ali zelena mrena, bolezen, ki prizadane očesni živec, z 2% vseh slabovidnosti [10].



Slika 3.1: Originalna slika [11]

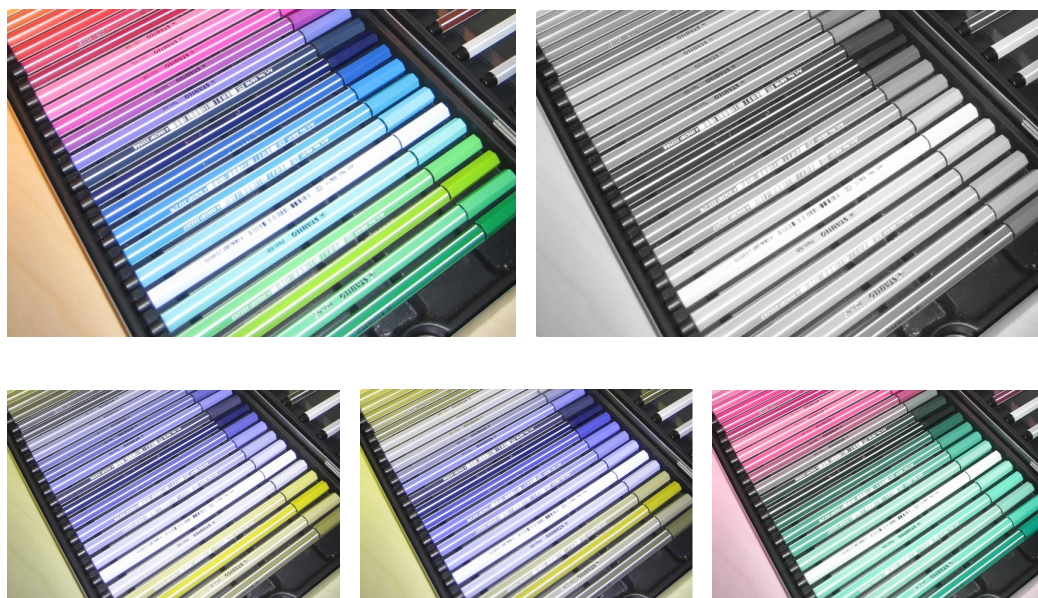


Slika 3.2: Primer vida s sivo mrežo [11]



Slika 3.3: Primer vida z glaukomom [12]

Posamezne okvare različno vplivajo na vid. Nekatere sliko zameglijo, druge zmanjšajo vidno polje ali ustvarijo zamaknjene in podvojene slike. Lahko jih spremlja tudi barvna slepota, nezmožnost razlikovanja med nekaterimi ali vsemi barvami. Na naslednjih slikah so ilustrirane najpogostejše oblike barvne slepote, narejene z uporabo Colorblind Design 1.1.4, vtičnika za spletni brskalnik Firefox.



Kot lahko opazimo, je z barvno slepoto težko razlikovati posamezne barve, sploh če gre za odtenke iste barve. Pri razvoju aplikacij za slabovidne je treba zato paziti, da je med elementi dovolj velik kontrast svetlo-temno, tako da jih lahko brez težav razlikujejo tudi ljudje, ki ne zaznavajo vseh barv.

3.2 Pripomočki za branje zaslona

Računalnik večino povratnih informacij posreduje skozi zaslon. Ker slabovidni težje zaznavajo z očmi, se je razvilo nekaj pripomočkov, ki jim olajšajo branje z zaslona. Mednje sodijo bralci zaslona, ki berejo kar je trenutno v fokusu, povečevalci zaslona, ki povečajo dele zaslona in Braillovi prikazovalniki, ki tekst v fokusu posredujejo na ploščico, iz katere se ga potem lahko prebere v obliki Braillove pisave.

Če želimo, da je naša aplikacija primerna tudi za slabovidne, moramo poskrbeti, da je kompatibilna z zgoraj naštetimi pripomočki. Paziti je treba predvsem na to, kateri elementi se kdaj fokusirajo in da aplikacija ne uporablja istih bližnic na tipkovnici, kot jih uporabljajo tudi ti pripomočki.

3.2.1 Povečevalci zaslona

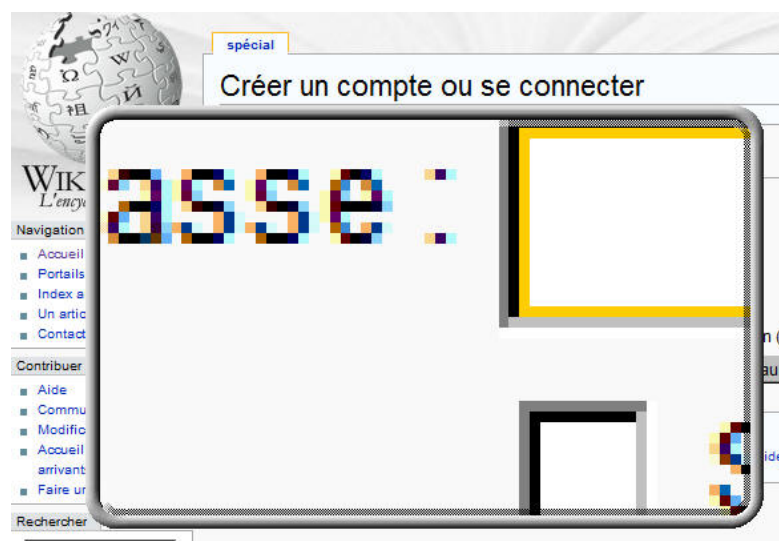
Povečevalec zaslona je program, ki poveča določen del zaslona za n -krat. Deluje kot pomoč tistim, ki sicer slabo vidijo, ampak lahko še vedno razločijo posamezne elemente, če so ti dovolj povečani. Navadno so sprogramirani tako, da sledijo kurzorjem (miškinem, v tekstu, ipd.), oziroma da sledijo fokusu, če se pomikamo s tipkovnico. Boljši povečevalci so tudi popolnoma nastavljivi in ponujajo veliko paleto funkcij:

- Barvna inverzija: Veliko slabovidnih lažje razlikuje belo na črnem kot črno na belem ali obratno. S pomočjo barvne inverzije lahko dosežemo željeno kombinacijo na kateremkoli tekstu.
- Glajenje teksta: Če nekajkrat povečamo tekst, postane ta popačen, kvadratast in težje prepoznaven. Povečevalci zaslonov zato uporabljajo algoritme, ki takšen tekst popravijo, da je na videz enak nepovečanemu.
- Prilagajanje kurzorja: Povečevalci omogočajo spremembo kurzorja, da ta čim bolj ustreza potrebam in željam uporabnika.
- Različne vrste povečave: Omogočajo, da se povečan del prikaže čez celoten zaslon, ali pa se le kot lupa pomika po zaslonu in poveča majhen delež naenkrat.
- Branje zaslona: Nekateri povečevalci delujejo tudi kot osnovni bralci zaslona in znajo prebrati povečan tekst.

Večina operacijskih sistemov ima že vgrajen enostaven povečevalec zaslona ali pa je ta dostopen zastoj iz njihovih spletnih trgovin. To velja tako za osebne računalnike kot tudi za mobilne telefone in tablične računalnike.

3.2.2 Bralci zaslona

Bralci zaslona delujejo podobno kot povečevalci zaslona s to razliko, da fokusiranega teksta ne povečajo, temveč ga preberejo in ga nato pošljejo skozi



Slika 3.4: Primer delovanja povečevalca zaslona [13].

zvočnike ali na Braillov prikazovalnik.

Bralci zaslonov so prav tako vgrajeni v večino operacijskih sistemov, tako na osebnih računalnikih, kot tudi na mobilnih napravah.

3.2.3 Braillovi prikazovalniki

Braillov prikazovalnik je naprava, ki jo lahko priklopiš na računalnik in nato prevaja tekst na zaslonu v Braillovo pisavo. Namesto vsake črke ima naprava 8 luknjic, ki predstavljajo eno črko v Braillovi pisavi. Ko želi prikazati posamezno črko, samo dvigne zaobljene dele skozi luknjice, tako da so na otip podobni knjigam v Braillovi pisavi.

3.3 Principi razvoja aplikacij za slepe in slabovidne

Pri razvoju aplikacij za slepe in slabovidne je potrebno upoštevati določene kriterije, katerim navadno ne posvečamo prevelike pozornosti. Kot je nave-



Slika 3.5: Primer Braillovega prikazovalnika [13].

deno na spletni strani AFB [9] je teh principov osem:

1. Uporabljaljaj dosledne standardne elemente uporabniških vmesnikov.

Naredi vmesnik, ki je dosleden skozi celo aplikacijo in podoben drugim že obstoječim vmesnikom, tako da se lahko uporabnik hitro nauči uporabljati program z veščinami, ki jih je pridobili pri uporabi drugih aplikacij. Ko je le mogoče, uporabi standardne elemente operacijskega sistema za prikaz orodne vrstice, menijev, ipd. Veliko bralcev zaslona je namreč zasnovanih tako, da prepoznajo te standardne komponente. Če so uporabljene neke druge komponente, se lahko zgodi, da bralec zaslona ne bo znal pretvoriti te komponente v govor, s čimer bo ta del aplikacije postal neuporaben.

2. Naredi vmesnik, ki je prilagodljiv

Dovoli uporabniku, da lahko vmesnik popolnoma prilagodi svojim željam in potrebam. Omogoči spremembo velikosti in tipa pisave, barv, kontrasta, dovoli spremembo stila kurzorja in zvočnih povratnih informacij. Možnost prilago-

ditve prikazane vsebine lahko pozitivno vpliva na vse uporabnike aplikacije, ne le na te s slabšim vidom.

3. Poskrbi, da je vmesnik popolnoma upravljiv z uporabo tipkovnice

Omogoči dostop do vseh funkcij in lastnosti programa samo z uporabo tipkovnice. Vsi vidiki programa, vključno z namestitvijo, naj bodo omogočeni brez uporabe miške. Pozoren je treba biti na to, da so programi, ki omogočajo slepim in slabovidnim lažjo uporabo računalnika, pogosto upravljani s tipkovnico, zato mora biti razvijani program sposoben delitve tipkovnice z drugimi programi.

4. Dodaj tekstovne opise vsem grafičnim elementom in ikonam

Poskrbi, da so vsem grafičnim elementom dodani tekstovni opisi, tako da se jih lahko pretvori v zvočno informacijo z bralci zaslona in se jih lahko pošlje v Braillove prikazovalnike. Prav tako je slabovidnim lažje prebrati tekst, kot prepoznati, kaj slika ali ikona predstavlja, obenem pa lahko tudi preostali uporabniki programa lažje prepoznajo, za kaj se neka ikona uporablja.

5. Uporabi standardne načine prikazovanja fokusa

Določi fokus z uporabo standardnih orodij, ki so del operacijskega sistema, kot je na primer miškina puščica. Bralci in povečevalci zaslona morajo biti vedno zmožni sledenja fokusu.

6. Ne zanašaj se na barvo, kot edini nosilec informacije

Zagotovi dodatna sredstva za posredovanje informacij. Navodila v obliki barv so lahko nevidna barvno slepim, slabovidnim in nedosegljiva Braillovim prikazovalnikom. Omogoči, da si lahko uporabnik nastavi svoje barve pisave in ozadja.

7. Onemogoči, da določene komponente izginejo po nekaj trenutkih

Novim računalniškim uporabnikom, uporabnikom podporne tehnologije in tistim, ki slabše razumejo jezik, lahko vzame več kot nekaj sekund ali minut, da najdejo in dojamejo stvari, kot so na primer opozorilno sporočilo ali namig o uporabi orodja.

8. Podpiraj dostopnost v namestitvi in konfiguraciji

Aplikacija se mora držati vseh zgornjih načel pri njeni namestitvi in konfiguraciji, tako da je lahko nameščena s strani uporabnika, ki uporablja podporne tehnologije, kot so bralec zaslona.

Poglavje 4

Pregled področja

Za učenje tipkanja obstaja že veliko tako namiznih kot tudi spletnih aplikacij. V naslednjih točkah si bomo ogledali nekaj spletnih, ker so te bolj zanimive zaradi sorodnosti z našo aplikacijo, pa tudi eno namizno.

4.1 TypingClub

TypingClub [1] je brezplačna spletna aplikacija, ki je namenjena učenju tipkanja in višanju njegove hitrosti. Ponuja 100 različnih vaj, od katerih so nekatere namenjene učenju tipkanja abecede, simbolov in števil, druge pa urjenju v hitrosti tipkanja. Vaje si sledijo v smiselnem zaporedju in šele ko uspešno opraviš vajo, se odklene naslednja. Aplikacija sicer omogoča tudi preskakovanje vaj, a to močno odsvetuje.

Pred opravljanjem prve vaje, se prikažejo navodila, ki vključujejo pravilno držo pred računalnikom in pravilno postavitev prstov. Med opravljanjem vaje je na dnu zaslona prikazana tipkovnica, ki kaže, na katero tipko je treba pritisniti v naslednjem koraku, prikazani pa so tudi prsti in katerega od njih je treba uporabiti v naslednjem koraku. Posameznega od teh dveh elementov

lahko po potrebi skrijemo.



Slika 4.1: Vaja črk D in K [1].

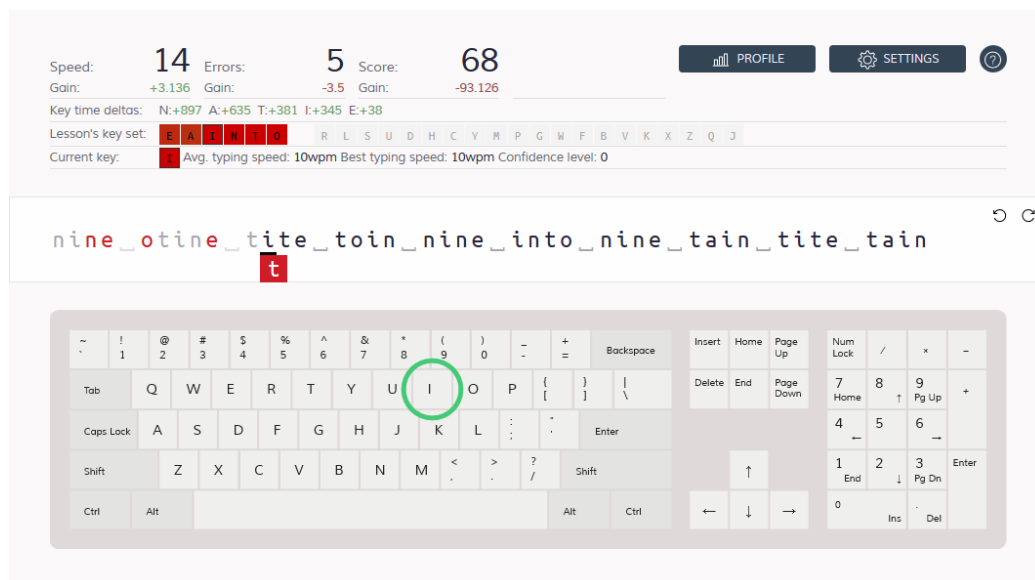
Ob tipkanju se sproti računata hitrost tipkanja v besedah na minuto in pravilnost tipkanja. V odsekih besedila, kjer ti gre še posebej dobro, te aplikacija tudi pohvali. Ob koncu tipkanja se pokaže tvoj rezultat. Če ti je šlo dobro, ti aplikacija ponudi možnost nadaljevanja na naslednjo vajo, sicer pa ti ponudi le možnost ponovnega opravljanja iste vaje.

Aplikacija se mi zdi dobra, ker je zelo pregledno urejena, vaje pa si sledijo v logičnem zaporedju od vaj za spoznavanje tipkovnice, do vaj, ki zahtevajo hitrost vsaj 45 besed na minuto. Pomankljivosti vidim v tem, da je

možna uporaba le ene vrste QWERTY tipkovnice, zato so namigi za slovensko QWERTZ večkrat napačni, sploh pri učenju simbolov. Poleg tega aplikacija ni najbolje prilagojena za uporabo slepih in slabovidnih, ker je kontrast med tekstom in ozadjem premajhen, barvnih shem ni mogoče spreminjati, prsti na tipkovnici pa prekrivajo tipke in kot namig le nekoliko spremenijo barvo obrobe. Ne nudi tudi nobene zvočne informacije.

4.2 keybr.com

Keybr.com [2] je prav tako kot TypingClub v 4.1 brezplačna spletna aplikacija, ki je namenjena učenju tipkanja in višanju njegove hitrosti. Za razliko od TypingCluba pa nima določenih vaj, temveč nam za začetek ponudi nekaj najpreprostejših črk na tipkovnici, katerih nabor se nato veča glede na uspešnosti uporabnika. Možno je tudi sestaviti svoj nabor črk, vendar brez števil in simbolov. Po potrebi lahko prikaže tudi navodila, ob izgubi fokusa pa se na tipkovnici za pomoč izriše pravilna postavitev prstov.



Slika 4.2: Osnovna vaja v keybr.com [2].

Ob tipkanju se sproti računajo hitrost tipkanja in napake, računa pa se tudi napredek skozi vajo in koliko časa v povprečju porabiš za pritisk posamezne tipke. Če si uporabnik ustvari profil, se ti podatki beležijo skozi čas. Ob dobrih rezultatih prejemaš tudi medalje.

Keybr.com je tudi veliko bolj prilagodljiv kot TypingClub iz 4.1. Uporabnik ima možnost uporabe pet različnih povečav in dveh barvnih shem (črno na belem in belo na črnem). Poleg tega je na voljo 13 različnih tipkovnic, med katerimi je tudi ena QWERTZ tipkovnica, ki pa je na žalost francoska in nima šumnikov. Lahko se tudi vklopi zvok, ki uporabnika opozori na napake.

Stran keybr.com je zelo pregledna, vsebuje pa tudi generator naključnih angleških besed iz nabora črk, kar naredi vaje tipkanja bolj smiselne in zanimive. Edini pomankljivosti sta ti, da ne vsebuje slovenske tipkovnice in da se na njej ne more vaditi tipkanja simbolov.

4.3 Stamina

Stamina [23] je brezplačna namizna aplikacija iz leta 2005, ki jo uporabljajo na ZSSM. Ker dobi tipkovnico iz operacijskega sistema, podpira tudi slovensko tipkovnico. Omogoča sestavljanje osnovnih vaj na podlagi tipkovnice, podpira pa tudi vključevanje lastnih vaj. Nudi ne nobenih začetnih navodil, ampak le tipkovnico z namigi.



Slika 4.3: Stamina [23].

Stamina je zanimiva za učence, ker vsebuje več zabavnih zvočnih posnetkov. Je tudi vizualno prilagodljiva in ob koncu iteracije računa uspešnost uporabnika, ki pa se ne hrani, temveč se takoj po ogledu izbriše.

Poglavje 5

Aplikacija za učenje tipkanja

5.1 Zahteve za aplikacijo

- *Spletna aplikacija:* Aplikacija naj bo del spletnega portala, ki hrani uporabnikove rezultate skozi čas in učiteljem ter učencem omogoča pregled napredka posameznika. Poleg tega naj bodo na istem mestu dostopna tudi navodila in napotki za reševanje vaj ter posamezne instance vaje. Uporabljene naj bodo spletne tehnologije (glej 5.2.1).
- *Prilagojena naj bo za uporabo slepih in slabovidnih:* Da je aplikacija prilagojena za uporabo slabovidnih je treba poskrbeti za uporabo ustreznih barvnih kontrastov in razločnih pisav ter za kompatibilnost s povečevalci in bralci zaslona. Uporaba miške naj ne bo potrebna (glej 3.3).
- *Prilagojena naj bo za otroke:* Aplikacija naj bo enostavna in simpatična, na koncu naj bo uporabnik nagrajen z zvezdicami.
- *Omogoča naj več različnih oblik prikazovanja teksta:* Ker ima vsak uporabnik svoje preference, naj ima možnost izbire več različnih vrst prikazovanja teksta in števila prikazanih znakov (glej 5.3.1).

- *Vključenih naj bo več različnih vaj:* Vključenih naj bo več instanc aplikacije, vsaka pa naj vključuje posamezno vajo. Instance naj si med seboj sledijo v smiselnem zaporedju.
- *Računanje statistike:* Ob koncu posamezne vaje naj aplikacija izračuna več kriterijev uspeha, poleg tega pa naj računa tudi uspešnost vsakih 30 sekund in jo na koncu predstavi v odvisnosti od časa (glej 5.3.5).
- *Prikaz pomoči:* Uporabnika naj aplikacija usmerja z napotki o tem, s katerim prstom je potrebno pritisniti na katero tipko na tipkovnici v naslednjem koraku vaje (glej 5.3.4). Po želji naj je zmožna tudi prikazati osnovnih navodil za slepo tipkanje (glej 5.3.3).

5.2 Tehnologije

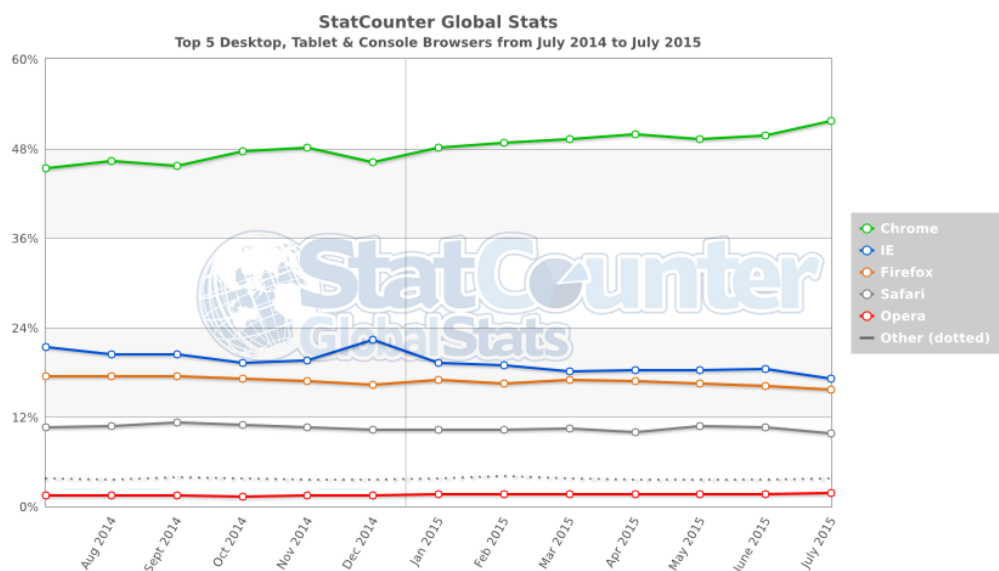
Kot sem že omenila, je bil cilj moje diplomske naloge izdelati aplikacijo, ki bo osnovnošolcem pomagala na poti do znanja slepega desetprstnega tipkanja. Ker smo se odločili, da jo vključimo v projekt ZaznajSpoznaj, ki je spletno-mobilna rešitev napisana v php-ju, je bilo treba tudi mojo aplikacijo sprogramirati v spletnih tehnologijah. Konkretno smo se odločili za uporabo JavaScripta, ker je ta zelo primeren za izdelavo front-end delov spletnih strani.

5.2.1 Spletno programiranje

Spletno programiranje je pojem, ki se uporablja za opis razvoja aplikacij, ki so namenjene za delovanje na internetu ali intranetu. Za izdelavo tovrstnih aplikacij navadno uporabljamo arhitekturo odjemalec/strežnik, pri kateri odjemalec zahteva storitev, strežnik pa se na zahteve odzove. Tipični odjemalci so brskalniki, ki običajno preko protokola HTTP komunicirajo s strežnikom, strežniki pa so programi, ki zagotavljajo dokumente spletnim odjemalcem.

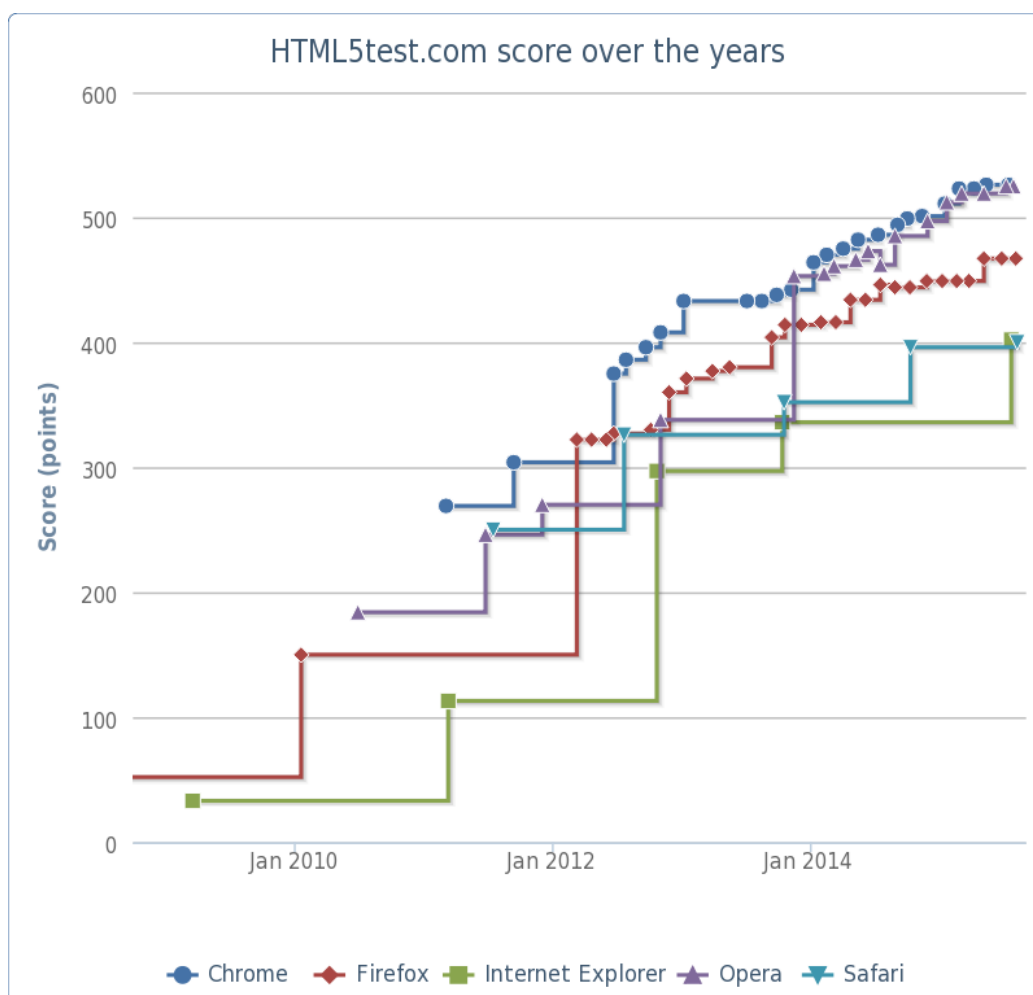
Brskalniki

Poznamo več različnih spletnih brskalnikov. Med največje sodijo Firefox (Mozilla), Internet Explorer (Microsoft), Google Chrome (Google), Opera (Opera Software) in Safari (Apple). En največjih izzivov, s katerimi se mora programer soočiti pri razvoju spletnih aplikacij, je prikazati spletno vsebino vizualno podobno in delujočo v čim večjem številu spletnih brskalnikov. Težava ni le v tem, da različni brskalniki temeljijo na različnih pogonih in torej prikazujejo posamezne elemente drugače, temveč tudi podpirajo različne verzije HTML in CSS. V praksi to pomeni, da se spletne strani, izdelane z uporabo funkcionalnosti, ki jih ponujata najnovejša HTML in CSS, ne bodo prikazale v vseh brskalniki, kaj šele v vseh starejših verzijah teh brskalnikov. Na srečo nam CSS omogoča, da izoliramo kritične odseke kode in jih nadomestimo s preprostejšimi.



Slika 5.1: Globalni odstotek uporabnikov posameznega brskalnika v zadnjem letu [19]

Pri razvoju prenosljive spletne aplikacije obstaja nenapisano pravilo, da začnemo razvoj v katerem od boljših brskalnikov, nato pa stran postopoma prilagajamo še za prikaz v nekaterih manj zmogljivih brskalnikih. Navadno se osredotočimo le na večje brskalnike z opaznim številom uporabnikov. Najtrši oreh predstavljajo starejše verzije Internet Explorerja, saj imajo te, kljub temu da so že nekoliko zastarele, še vedno kar veliko število uporabnikov.



Slika 5.2: Doseženi rezultati na html5test.com v zadnjih letih [20].

5.2.2 JavaScript in uporabljene knjižnice

Ker moja aplikacija skoraj v celoti deluje na strani odjemalca, sem zanjo uporabila JavaScript. JavaScript je namreč programski jezik, ki omogoča, da skripte na strani odjemalca komunicirajo z uporabnikom in spreminjajo vsebino strani, ko je ta že prikazana. Pri razvoju sem si pomagala tudi s sledečimi knjižnicami:

- jQuery [24]: jQuery je JavaScript knjižnica namenjena poenostavitvi programiranja na strani odjemalca. Med drugim poenostavlja navigacijo po dokumentu, izbiro elementov DOM, obravnavanje dogodkov, delo s CSS, rokovanje z dogodki, ustvarjanje animacij in efektov ter razvoj AJAX aplikacij.
- jQuery UI [25] jQuery UI je razširitev knjižnice jQuery, ki vsebuje dodatne elemente za prikaz in interakcijo z uporabnikom, kot so na primer izboljšani gumbi, dialogi in zavihti. V programu za učenje tipkanja sem iz te knjižnice uporabila drsnik, ki na koncu igre vizualizira igralčev uspeh.
- CanvasJS [26]: CanvasJS je JavaScript knjižnica, ki je namenjena vizualizaciji statističnih podatkov z grafi.
- Color Picker [27]: colorpicker.js je še ena knjižnica, ki temelji na jQueryju in uporabniku omogoča preglednejši izbor barv.

5.2.3 Prilagoditev spletnih aplikacij za uporabo pripomočkov za branje zaslona

Veliko spletnih mest uporablja skripte, ki posodablajo le nekatere dele strani in njihovo vsebino, ne da bi ob tem ponovno nalagale celotno spletno stran. To lahko ustvari probleme za slepe in slabovidne, saj pripomočki za branje zaslona v osnovi ne vedo, kaj posamezen del predstavlja. Ob tem lahko pride do tega, da do uporabnika ne pridejo pomembne informacije, ki mu jih želi

stran sporočiti, temveč le kup nepomembnih. Leta 2014 so zato W3C izdali specifikacijo WAI-ARIA [17], ki opisuje kako zasnovati spletno mesto, da bo čim bolj dostopno vsem uporabnikom. WAI-ARIA omogoča, da pripomočki za branje zaslona ne vidijo strani kot statičen dokument, temveč ga prepoznajo kot aplikacijo. Stranem namreč narekuje, da posameznim elementom dodajo vlogo, lastnosti, stanje in prioriteto, kar omogoča pripomočkom za branje zaslona učinkovito in smiselno branje. WAI-ARIA predstavlja nabor značk, ki jih HTML elementom lahko dodamo tako:

```
<span role="status" aria-level="1">tekst</span>
```

V tem primeru bodo pripomočki za branje zaslona element razumeli kot status in ga uporabniku sporočili kot štatus tekst". WAI-ARIA podpirajo vsi večji brskalniki, razume pa ga večina bralcev zaslona.

Tudi z uporabo WAI-ARIA igra fokus v branju aplikacij še vedno veliko vlogo, še posebno če želimo programsko menjati elemente v fokusu. Posebna funkcija za to je v JavaScriptu `.focus()`, toda z uporabo le te funkcije bralci zaslona elementa ne preberejo takoj. To dosežemo tako, da elementu nastavimo negativni `tabindex` in ga s tem izključimo iz vrstnega reda prikazovanja elementov:

```
$( '#id1' ).attr( 'tabindex', -1 );  
$( '#id1' ).focus();
```

5.2.4 Uporaba SVG slik v spletnih aplikacijah

Pri aplikacijah za učenje tipkanja je priporočljivo, da realiziramo pomoč, ki uporabniku narekuje, kje na tipkovnici se nahaja posamezna tipka in s katerim prstom moramo pritisniti nanjo. Če bi želeli imeti za vsako tipko v pomoč eno sliko, bi morali vključiti po šest slik za vsako roko in več kot 50 slik za tipkovnico, kar je prostorsko in časovno potratno. Da se slika naloži v HTML namreč traja nekaj časa, kar bi ob hitrem spreminjanju prikazanih slik pomenilo nepregledno utripanje slik. Zato smo se odločili za uporabo

SVG formata in zmanjšali število potrebnih slik na tri.

SVG je vektorski format dvodimenzionalnih slik, ki temelji na XML-u. Za nas je bil zanimiv predvsem zato, ker podpira interaktivnost, XML datoteke lahko namreč zlahka preiskujemo in manipuliramo. Poleg tega se kvaliteta slike v SVG ohranja ne glede na povečavo, saj je slika zapisana kot skupek vektorjev, ki jih preprosto pomnožimo s faktorjem povečave. SVG v določeni meri podpirajo vsi večji moderni brskalniki.

Spodnji del SVG dokumenta predstavlja `<g>` značko, ki vsebuje zapis za izris take zvezde. Kot je razvidno iz zapisa, so SVG dokumeti zelo pregledni, določamo pa jim lahko tudi id, ki zelo poenostavi iskanje elementa po dokumentu in njegovo manipulacijo.

```
<g id="zvezda"><path fill="#FFED00" d="M41.827, 19.163
c-0.82, 0.042-1.668-0.572-1.883-1.365c0, 0
-4.813-17.805-9.133-17.798 c-4.309, 0.003-9.098,
17.814-9.098, 17.814c-0.214, 0.794-1.061, 1.402-1.881
, 1.351c0, 0-18.334-1.144-19.742,3.202 c-1.41, 4.346,
14.127, 14.172, 14.127, 14.172c0.695, 0.439, 1.033,
1.43, 0.751, 2.202c0, 0-6.354, 17.399-3.097, 19.759
c3.262, 2.363, 17.817-9.084, 17.817-9.084c0.646-0.509
, 1.709-0.516, 2.362-0.017c0, 0,14.768, 11.287,
17.83, 9.067
c3.051-2.229-3.153-19.75-3.153-19.75c-0.274-0.775,
0.063-1.777, 0.753-2.227c0, 0, 15.438-10.084,14.104
-14.186 C60.247, 18.208, 41.827, 19.163, 41.827,
19.163z"/>
</g>
```

V spodnjih podpoglavjih si bomo ogledali, kako vključiti SVG slike v HTML in jih nato do njih dostopati z JavaScriptom.



HTML

Če želimo SVG elemente manipulirati, je za njihovo vključitev v HTML najprimernejša značka `<object>`, namenjena vključevanju multimedijskih vsebin v HTML. Ker nekatere starejše verzije brskalnikov ne podpirajo SVG, naj bo med začetno in končno značko tekst, ki uporabnika obvesti o tem, da njegov brskalnik ne podpira SVG slik in da se tam nahaja manjkajoča vsebina. Poleg tega naj bo vsaj en od atributov `'type'` in `'data'` izpolnjen. Izpolnjena značka naj zgleda nekako tako:

```
<object id="tipkovnica" type="image/svg+xml" data="t.svg">
  Tvoj brskalnik ne podpira SVG slik
</object>
```

JavaScript

Do v HTML vključenega SVG dokumenta lahko nato iz JavaScripta dostopamo tako:

```
var a = document.getElementById("tipkovnica");
var svgDoc = a.contentDocument;
```

Ko imamo spremenljivko, ki hrani vsebino SVG dokumenta, lahko začnemo z manipulacijo. Do posameznega elementa enostavno dostopamo, če poznamo njegov id:

```
var tipka = svgDoc.getElementById("preslednica");
```

Elementu lahko nato spremenjamo lastnosti. Če želimo spremeniti vrsti red prikazovanja elementov moramo upoštevati pa to, da so elementi na začetku

SVG dokumenta izrisani v ozadju, bolj kot se pomikamo proti koncu dokumenta, bolj pa so ti v ospredju. Zato vrstni red izrisa preprosto spremenimo tako, da željeni element enostavno pomaknemo gor ali dol po dokumentu.

5.3 Rešitev

lasala kača jasa safalada sladkala safalada lasala časa
kad jafa ajda saj lak čakal lak časa čaka lak skakal kaja



Slika 5.3: Najosnovnejša vaja črk ASDFJKLČ.

Rezultat diplomske naloge je aplikacija za učenje tipkanja, prilagojena za najmlajše in ljudi, ki imajo težave z vidom. Možna je popolna prilagoditev vmesnika od barv do velikosti pisave, uporaba miške je neobvezna, poleg vidne povratne informacije pa aplikacija omogoča tudi zvočno.

Ker se hitrega tipkanja najlažje naučimo kar s tipkanjem, je aplikacija zasnovana tako, da nam ponudi tekst, ki ga moramo čim hitreje ter s čim manj napakami natipkati. Tipka, ki jo moramo natipkati je obrobljena, nanjo je postavljen fokus, aplikacija pa jo tudi zvočno imenuje. Ko pritisnemo na tipko, lahko sledita dve akciji:

- *Tipka je pravilna:* prej obrobljena tipka izgubi obrobo in se obarva v

določeno barvo (privzeto rumeno)

- *Tipka je napačna:* aplikacija nas zvočno obvesti o napaki, prej obrobljena tipka izgubi obrobo in se obarva v določeno barvo (privzeto rdečo)



Slika 5.4: Spremembe teksta ob pravilnem in napačnem tipkanju

Obrobo in fokus dobi nato naslednja tipka, prav tako pa nam aplikacija zvočno sporoči, katera tipka je to.

Ko natipkamo celotno prikazano besedilo, se zasliši simpatična glasba in pojavi se okno, na katerem je prikazano, kako dobro nam je šlo. Nato imamo dve možnosti: lahko ponovimo vajo ali nadaljujemo na naslednjo.



Slika 5.5: Rezultat ob koncu tipkanja

5.3.1 Načini prikazovanja teksta

Ker ima vsak posameznik različne preference glede tega, kako je tekst prikazan, smo se odločili, da v aplikacijo vključimo pet različnih možnosti:

1. *Prikaz vsega teksta naenkrat*: prikaže celotno generirano besedilo naenkrat, ne glede na to, kako dolgo je.
2. *Prikaz vrstice teksta*: prikaže vrstico teksta dolžine n znakov, pri čemer je n določen s strani uporabnika. Prikazana vrstica se zamenja, ko jo uporabnik natipka do konca.
3. *Prikaz vrstice teksta z zamikanjem*: prikaže vrstico teksta dolžine n znakov, pri čemer je n določen s strani uporabnika. Črka, ki jo je treba v posamezni iteraciji natipkati, je vedno na istem mestu, ker se ob vsaki iteraciji črke v nizu zamaknejo za ena v levo. Prva črka tako pade iz niza, na konec pa se postavi nova črka.
4. *Prikaz ene besede*: deluje podobno kot *prikaz vrstice teksta*, le da je v tem načinu vedno prikazana le ena beseda.
5. *Prikaz ene črke*: prikaže le posamezno črko teksta. Ko črko natipkamo, se vsebina vrstice zamenja šele, ko ponovno pritisnemo na poljubno tipko. Ta način ignorira presledke v generiranem besedilu.

5.3.2 Parametri

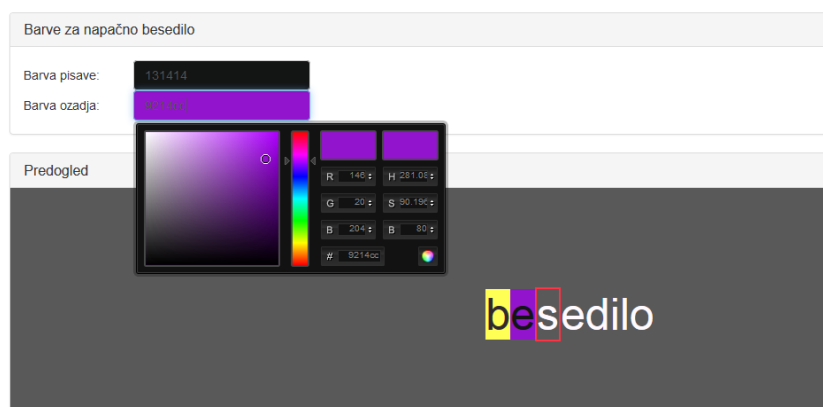
Da bi bila aplikacija čim bolj prilagodljiva smo vključili 13 nastavljivih parametrov:

1. *Tekst*: Vsaka instanca igre lahko vsebuje drugačno besedilo, kar posledično omogoča vajo tipkanja različnih črk in znakov. Kot vhod se v aplikacijo pošlje niz besed, ki se jih nato naključno premeša in iz njih vsakič ustvari novo besedilo.

2. *Način prikaza besedila*: Aplikacija omogoča pet različnih prikazov besedila, kot je opisano v 5.3.1. S tem parametrom nastavimo, kateri način želimo uporabiti.
3. *Število besed*: Ta parameter pove, koliko besed naj vsebuje besedilo, ki se generira iz vhodnega teksta.
4. *Dolžina vrstice*: Nekateri načini prikaza besedila potrebujejo ta podatek, da vedo, koliko znakov naj bo prikazanih v eni vrstici.
5. *Zahtevnost*: Uspešnost tipkanja se lahko na koncu računa na dva načina. En je namenjen na začetnike, drugi pa za tiste, ki so tipkanja že nekoliko vešči.
6. *Velikost pisave*
7. *Barva obrobe črke*
8. *Barva pisave*
9. *Barva pisave pravilno*
10. *Barva pisave napačno*
11. *Barva ozadja*
12. *Barva ozadja pravilno*
13. *Barva ozadja napačno*

Oznaki pravilno in napačno se tu nanašajo na barve že natipkanega besedila.

Za nastavljanje teh parametrov smo izdelali tudi urejevalnik, ki nudi pregleden izbor barv.



Slika 5.6: Del urejevalnika

5.3.3 Osnovna navodila za tipkanje

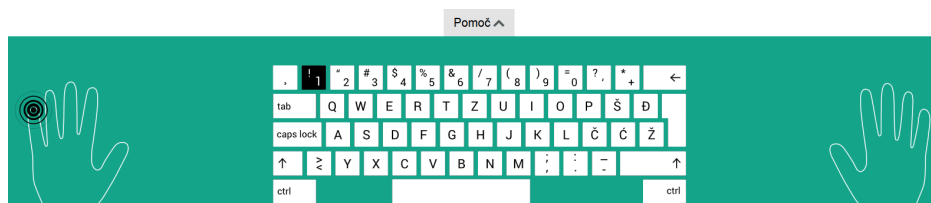
Uporabnik ima kadarkoli možnost prikaza osnovnih navodil za tipkanje, ki so jih napisali na ZSSM. Sestavljena so iz štirih podpoglavij. Prvo na hitro opiše pravilno držo pred računalnikom, drugo pove, s katerim prstom pritisnemo na katero tipko, tretje uporabnika opozori na pomen redne vaje, četrto pa narekuje pravilno zaporedje vaj.

5.3.4 Pomoč

Poleg zvočne pomoči nam aplikacija tudi vizualno pomaga, da pritisnemo pravo tipko. Na dnu zaslona se namreč nahaja tipkovnica in obe roki. Na tipkovnici je označena tipka, na katero moramo pritisniti, označen pa je tudi prst, s katerim moramo pritisniti na to tipko. Za tipkovnico smo uporabili eno SVG sliko, nato pa smo njene elemente ustrezno pomikali v ospredje, kot je opisano v 5.2.4. Po želji lahko pomoč skrijemo.

5.3.5 Statistika

Skozi vsako iteracijo aplikacija meri uporabnikov čas in beleži napake. Na koncu se izpiše število natipkanih znakov in porabljen čas. Izračunajo in izrišejo se še naslednji statistični podatki (za besedo se smatra 5 znakov):



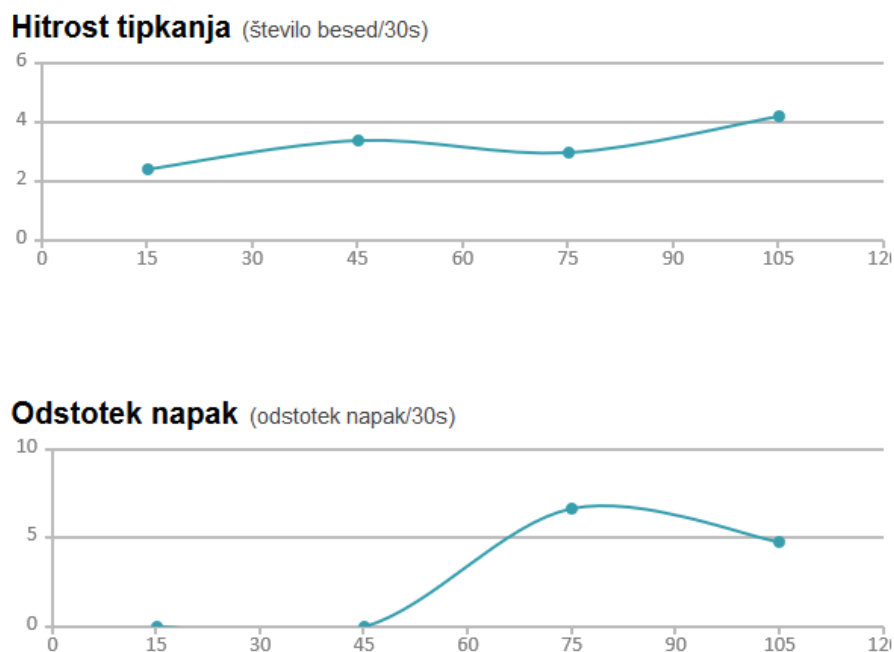
Slika 5.7: Primer za zahtevan pritisk na '!’.

- *Hitrost tipkanja*: beležena kot število besed na minuto.
- *Hitrost tipkanja z upoštevanjem napak*: beležena kot število besed na minuto, pri čemer se številu vseh natipkanih znakov odšteje večkratnik števila napak (pri osnovni težavnosti petkratnik, pri zahtevni pa desetkratnik).
- *Pravilnost*: beležena kot odstotek pravilno natipkanih znakov.
- *Graf hitrosti tipkanja*: vsakih 30 sekund beleži hitrost tipkanja v zadnjih 30 sekundah.
- *Graf hitrosti tipkanja z upoštevanjem napak*: vsakih 30 sekund beleži hitrost tipkanja v zadnjih 30 sekundah, a vsem natipkanim znakom odšteje večkratnik napačno natipkanih.
- *Graf odstotkov napak*: vsakih 30 sekund beleži odstotek napak v zadnjih 30 sekundah.

5.3.6 Ostale bližnjice

Na tipkah od F9 do F12 se nahajajo še nekatere uporabne funkcije. Te tipke smo si izbrali, ker jih pripomočki za branje zaslona navadno ne zasedajo:

- *F9*: Ob pritisku na to tipko se program ustavi in prikažejo se osnovna navodila za tipkanje. Ob ponovnem pritisku na tipko nadaljujemo s tipkanjem (glej 5.3.3).



Slika 5.8: Grafa za hitrost tipkanja in odstotek napak

- *F10*: Je gumb za vklop ali izklop zvočne pomoči, ki sporoča, na katero tipko moramo pritisniti.
- *F11*: Je gumb za vklop ali izklop zvočnih efektov, kot so pritisk na napačno tipko.
- *F12*: Ob pritisku na to tipko začasno ustavimo štetje časa, ki se nato ob ponovnem pritisku na tipko nadaljuje.

5.3.7 Sklopi vaj za tipkanje

Predvidenih je 11 sklopov vaj za tipkanje, ki si sledijo od prav osnovnih, pri katerih ni potrebno premikati prstov iz izhodiščne lege, do tistih nekoliko težjih. Vsi sklopi vsebujejo tudi črke njihovih predhodnikov.

1. ASDF JKLČ
2. GH
3. EI
4. RU
5. VM
6. C in vejica
7. OP
8. BN
9. TZ
10. Š in pika
11. Ž

Učitelj lahko po želji doda tudi instance aplikacije z drugimi znaki, kot so na primer simboli in velike začetnice.

Poglavje 6

Sklepne ugotovitve

Aplikacija za učenje desetprstnega tipkanja je nastala v sodelovanju z ZSSM in ekipo projekta ZaznajSpoznej. S skupnimi močmi smo poskušali izdelati program, ki osnovnošolcem in predšolskim otrokom na zanimiv način predstavi desetprstno tipkanje ter jih nato na poti učenja tipkanja usmerja in spodbuja. Potrudili smo se tudi, da bi bila aplikacija čim bolj prilagojena za uporabo slepih in slabovidnih uporabnikov, tako da smo implementirali kompatibilnost z bralci zaslonov, poskrbeli za velik kontrast med posameznimi elementi strani in omogočili personalizacijo barv.

Aplikacija lokalno deluje, smo pa ob vključevanju v spletni portal naleteli na nekaj težav, ki še niso popolnoma odpravljene. V naslednjih tednih tako nameravamo še popraviti izgled strani, saj se je CSS nekoliko podrl, in odpraviti manjše hrošče, ki so se pojavili v skripti. Realizirati moramo tudi še smiselno preklapljanje med posameznimi vajami. Poleg tega bi lahko v aplikacijo vgradili še kakšne dodatne funkcije, kot je na primer način tipkanja, pri katerem se program ob napačno natipkani črki ustavi in ne nadaljuje, dokler uporabnik ne pritisne prave črke, ali pa način pri katerem nismo znakovno, temveč časovno omejeni.

Veseli me, da nam je v določeni meri uspelo doseči vse zastavljene cilje in

upam, da bo program za učenje tipkanja učencem všeč ter da ga bodo tudi v praksi uporabljali. Razvoj aplikacije je bila sicer zelo prijetna izkušnja, skozi katero sem pridobila ogromno pozitivnih izkušenj in novega znanja.

Literatura

- [1] TypingClub [Online]. Dosegljivo:
<https://www.typingclub.com/>. [Dostopano 20. 8. 2015].
- [2] keybr.com [Online]. Dosegljivo:
<http://www.keybr.com/>. [Dostopano 20. 8. 2015].
- [3] A. Dobson. *Touch Typing in Ten Hours*. How to Content, 2002.
- [4] Keyboarding and Touch Typing for Children [Online]. Dosegljivo:
[http://www.abilitynet.org.uk/sites/abilitynet.org.uk/files/Keyboarding
%20and%20Touch%20Typing%20for%20Children.pdf](http://www.abilitynet.org.uk/sites/abilitynet.org.uk/files/Keyboarding%20and%20Touch%20Typing%20for%20Children.pdf). [Dostopano 20.
8. 2015].
- [5] Projekt ZaznajSpoznaj [Online]. Doseglivo:
<http://errorage.com/zs/>. [Dostopano 15.8.2015].
- [6] Kako sedeti? [Online]. Doseglivo:
http://www.mojmikro.si/v_srediscu/razkritje/kako_sedeti/. [Dostopano
15.8.2015].
- [7] Touch Typing Techique [Online]. Doseglivo:
<http://www.rapidtyping.com/typing-instructions.html/>. [Dostopano
15.8.2015].
- [8] 8 Benefits of Touch Typing Skills [Online]. Doseglivo:
[https://adamfortgo.wordpress.com/2014/06/25/benefits-of-touch-
typing-skills/](https://adamfortgo.wordpress.com/2014/06/25/benefits-of-touch-typing-skills/). [Dostopano 15.8.2015].

-
- [9] American Foundation for the Blind. *Creating Accessible Computer Applications* [Online]. Doseglivo:
<http://www.afb.org/info/accessibility/creating-accessible-computer-applications/25/>. [Dostopano 15.8.2015].
- [10] World Health Organization. *Visual impairment and blindness* [Online]. Doseglivo:
<http://www.who.int/mediacentre/factsheets/fs282/en/>. [Dostopano 20.8.2015].
- [11] Cataract [Online]. Doseglivo:
<https://en.wikipedia.org/wiki/Cataract/>. [Dostopano 11.8.2015].
- [12] Glaucoma. *Visual impairment and blindness* [Online]. Doseglivo:
<https://en.wikipedia.org/wiki/Glaucoma/>. [Dostopano 11.8.2015].
- [13] Wikipedia: Screen reader [Online]. Doseglivo:
https://en.wikipedia.org/wiki/Screen_reader/. [Dostopano 11.8.2015].
- [14] Five Best Browsers [Online]. Doseglivo:
<http://lifehacker.com/5178564/five-best-web-browsers/>. [Dostopano 11.8.2015].
- [15] How We Develop for Multiple Browsers [Online]. Doseglivo:
<http://oldtownmediainc.com/how-we-develop-for-multiple-browsers/>. [Dostopano 11.8.2015].
- [16] Optimizing content for different browsers: the RIGHT way [Online]. Doseglivo:
https://www.w3.org/community/webed/wiki/Optimizing_content_for_different_browsers:_the_RIGHT_way/. [Dostopano 11.8.2015].
- [17] WAI-ARIA [Online]. Doseglivo:
<http://www.w3.org/WAI/intro/aria/>. [Dostopano 11.8.2015].

-
- [18] Wikipedia: SVG [Online]. Doseglivo:
https://en.wikipedia.org/wiki/Scalable_Vector_Graphics/. [Dostopano 11.8.2015].
- [19] StatCounter [Online]. Doseglivo:
<http://gs.statcounter.com/>. [Dostopano 11.8.2015].
- [20] HTML 5 TEST [Online]. Doseglivo:
<https://html5test.com/>. [Dostopano 11.8.2015].
- [21] J. Duckett. *JavaScript and JQuery: Interactive Front-End Web Development*. Wiley, 2014.
- [22] J. Duckett. *HTML and CSS: Design and Build Websites*. Wiley, 2011.
- [23] Stamina [Online]. Doseglivo:
<http://typingsoft.com/overview.htm/>. [Dostopano 25.8.2015].
- [24] jQuery UI [Online]. Doseglivo:
<https://jquery.com/>. [Dostopano 11.8.2015].
- [25] jQuery UI [Online]. Doseglivo:
<https://jqueryui.com/>. [Dostopano 11.8.2015].
- [26] CanvasJS [Online]. Doseglivo:
<https://canvasjs.com//>. [Dostopano 11.8.2015].
- [27] Color Picker [Online]. Doseglivo:
<http://www.eyecon.ro/colorpicker/>. [Dostopano 11.8.2015].